
Noury
Bouraqadi
Dave
Mason
Richard
Ronteltap
Erik
Stel

ESUG 2025, Gdańsk, Poland

Smalltalk for the **Web**

& Beyond

Noury

Dave

Richard

Erik

WebST

PharJS

IMT Nord Europe
École Mines-Télécom
IMT-Université de Lille

learn, code, make
nootrix

Toronto Metropolitan University

SmallJS

UvA

CodeParadise

Smalltalk & the Web

New to Smalltalk?

Less than 2 years

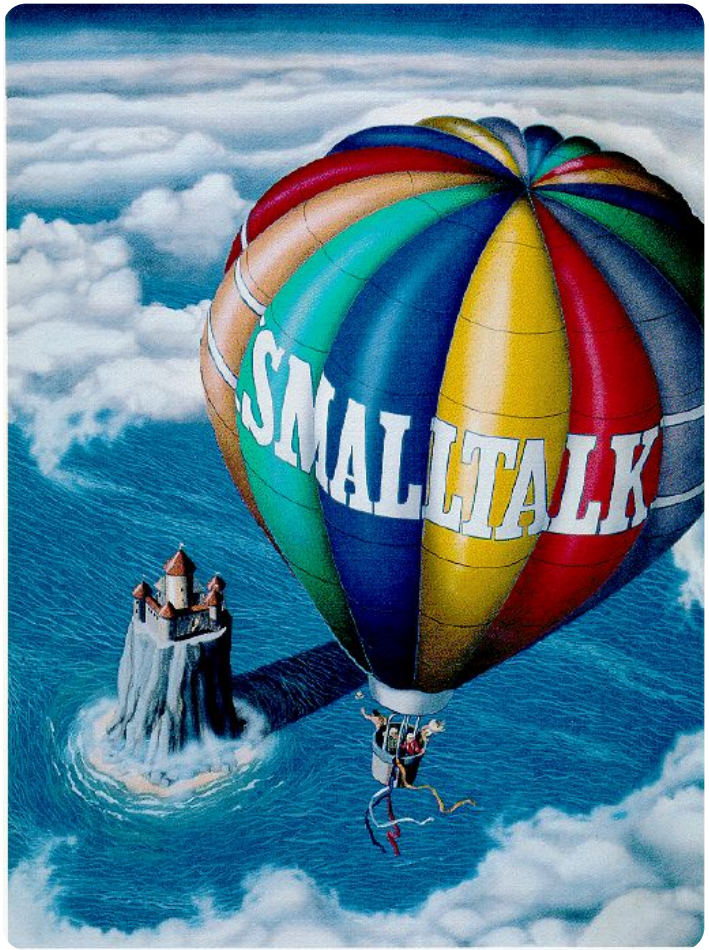
Smalltalk is dangerous.
It is a drug.

My advice to you would be

don’t try it.

It could ruin your life

– Andy Bower, CEO of Object Arts Ltd.



= Great Language, Libraries, Tools, Community


Smalltalk

Smalltalk
All the Time

Smalltalk All the Time

Software Lifecycle

Smalltalk Everywhere

Smalltalk Everywhere

On All Platforms

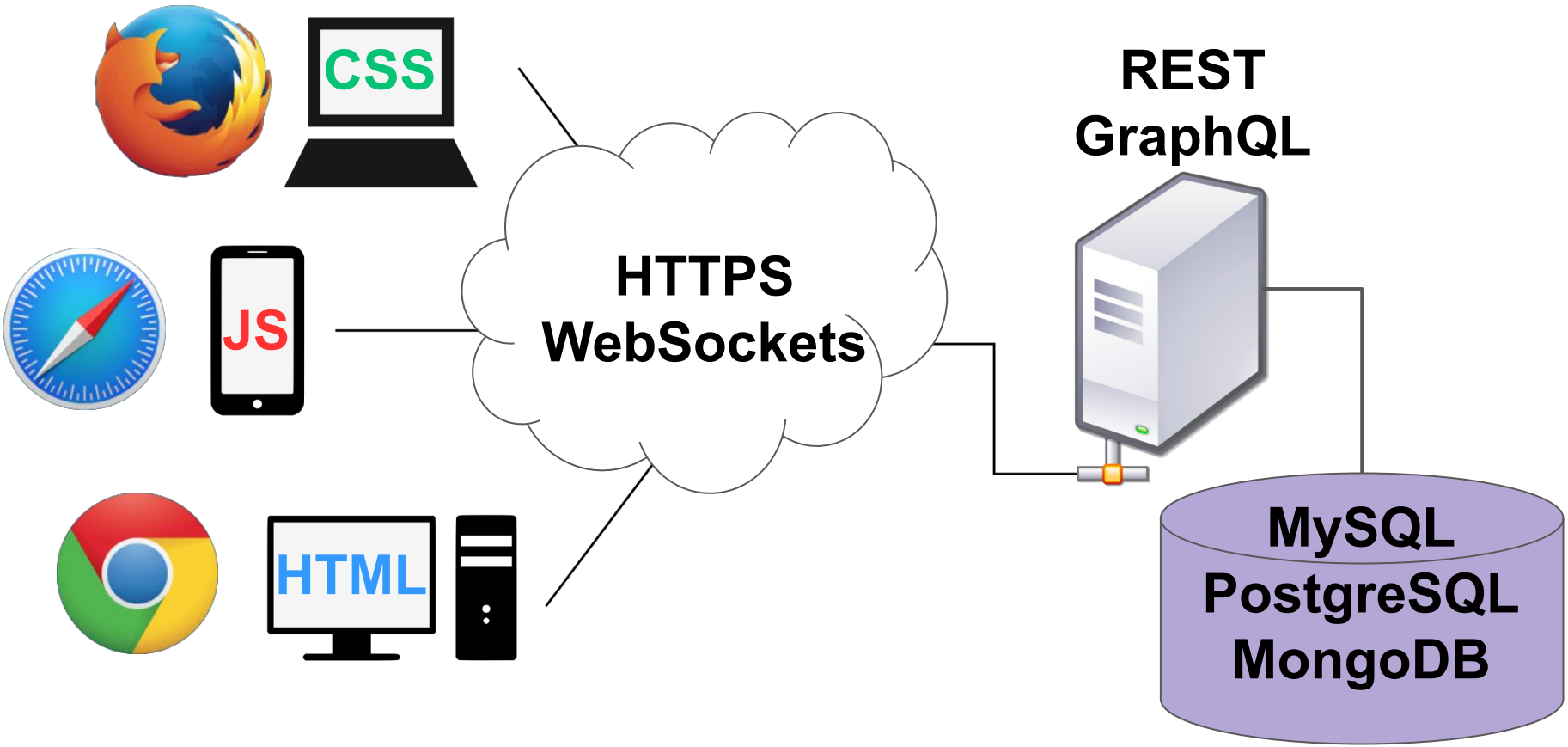
Smalltalk Everywhere

The **Web**

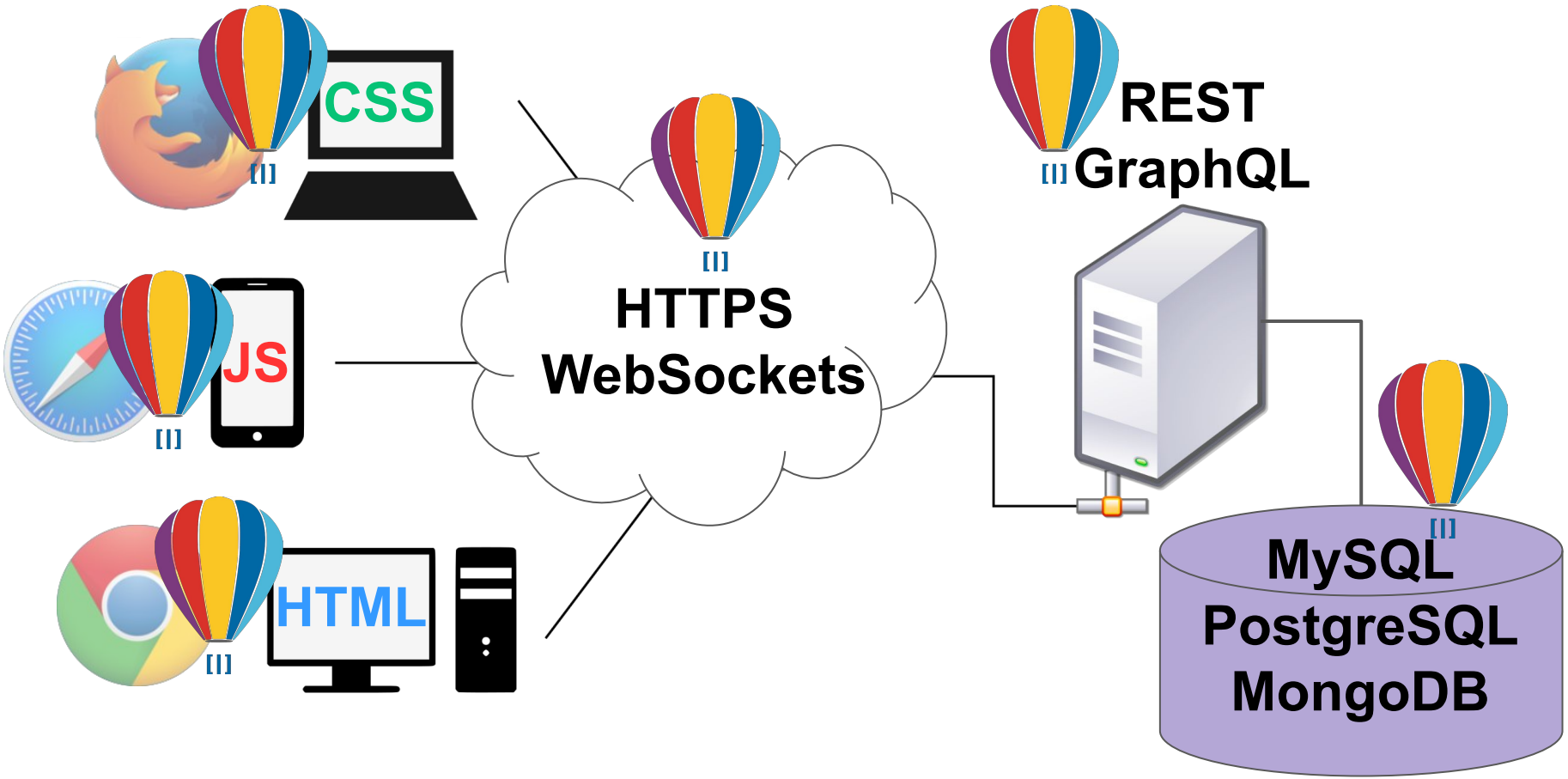
Are you in a

Web Project?

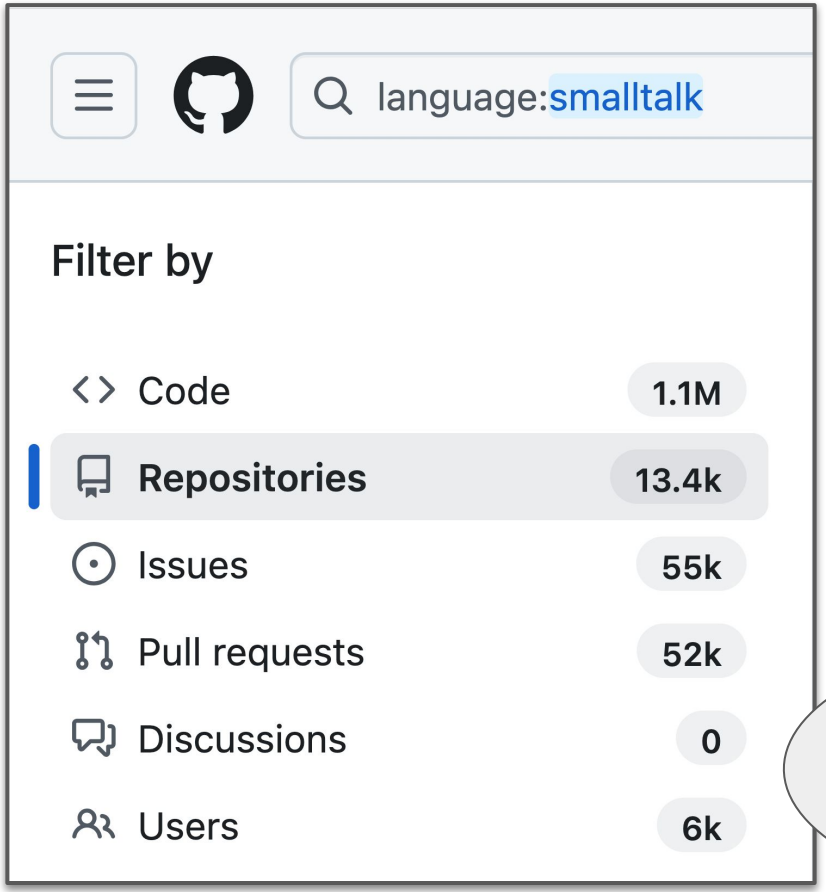
Web Technologies



Smalltalk for the Web

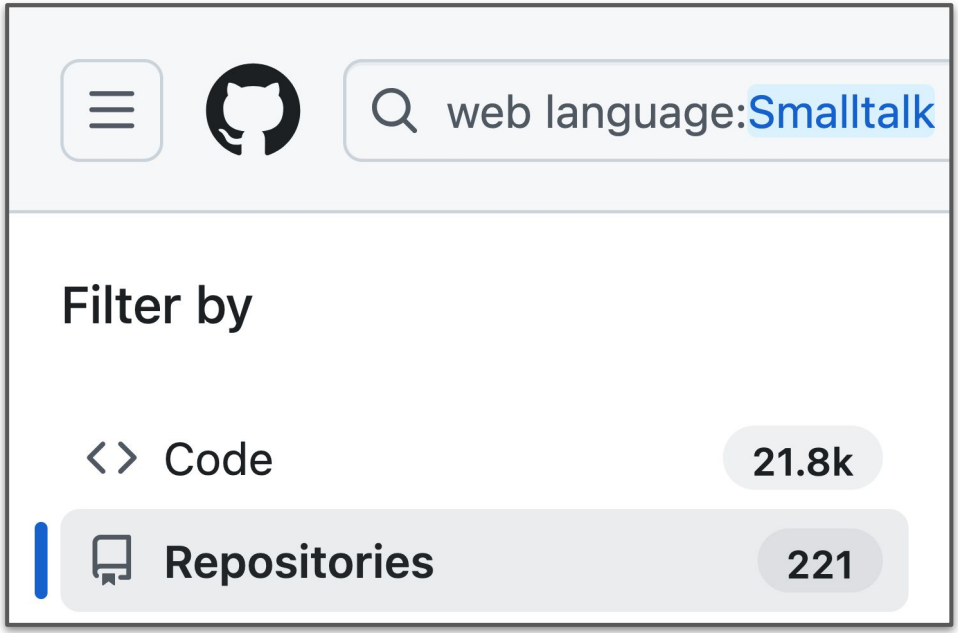


13.4K Github Repositories use Smalltalk

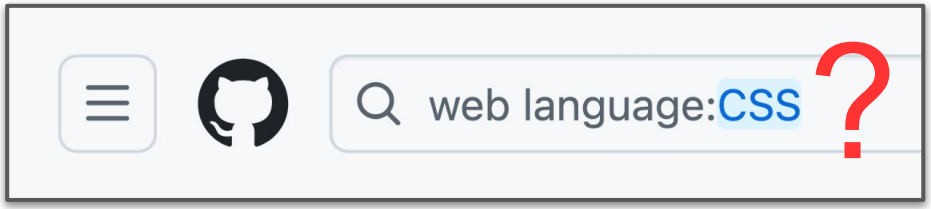


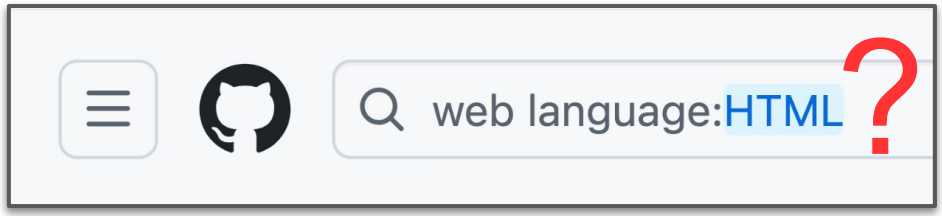
Developers
Responsibility to
Add Language Tags

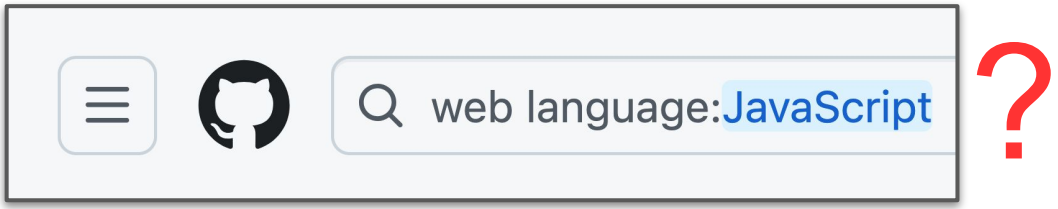
“Web” in 221 Smalltalk Repositories out of 13.4K

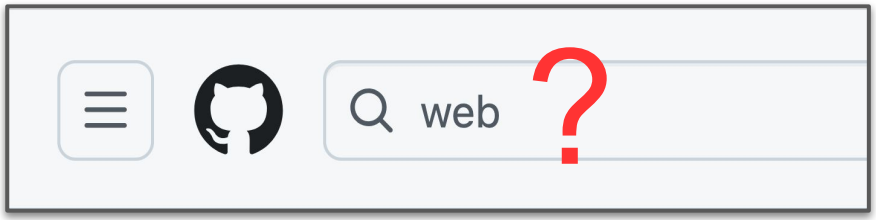


1.65%

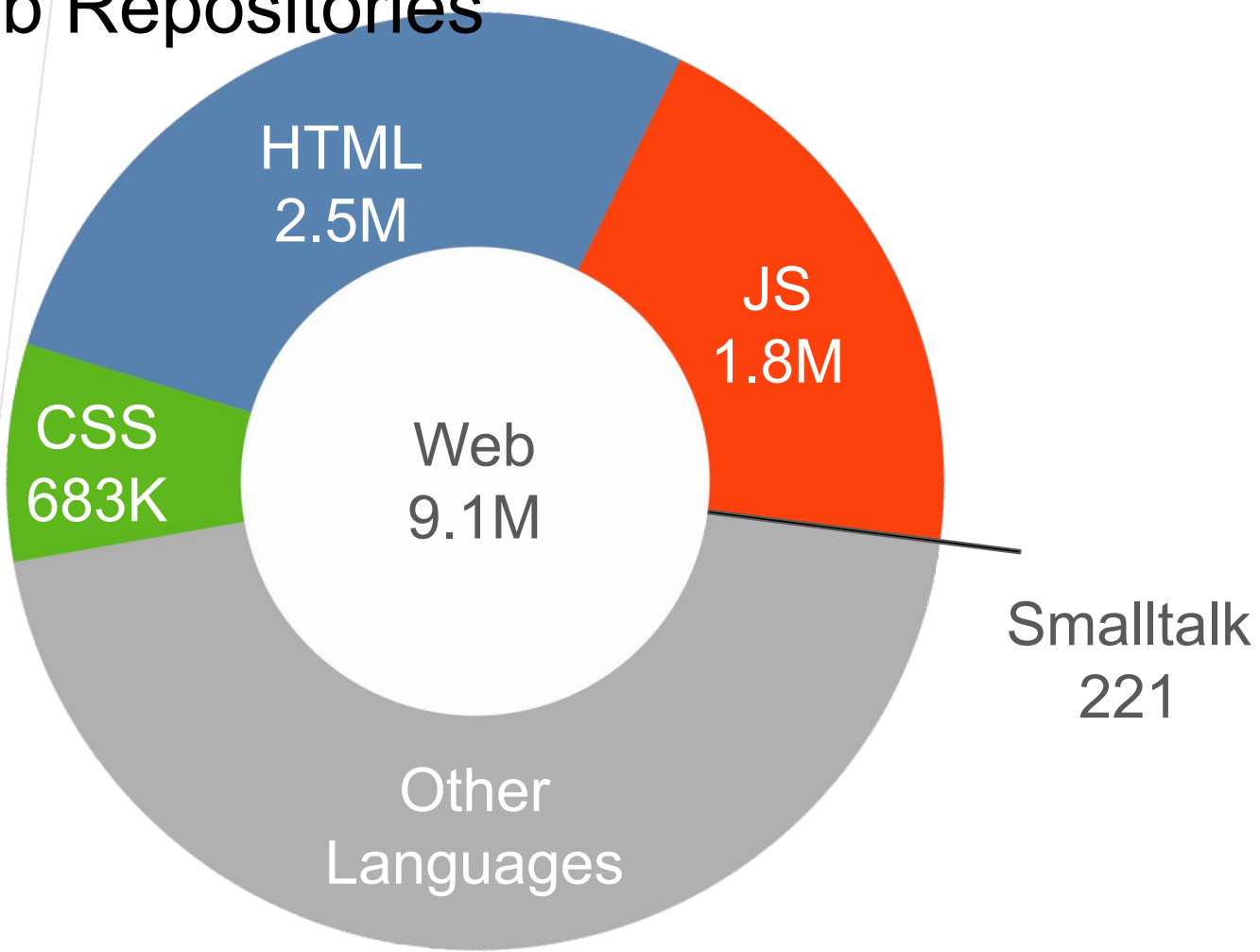

web language:CSS


web language:HTML

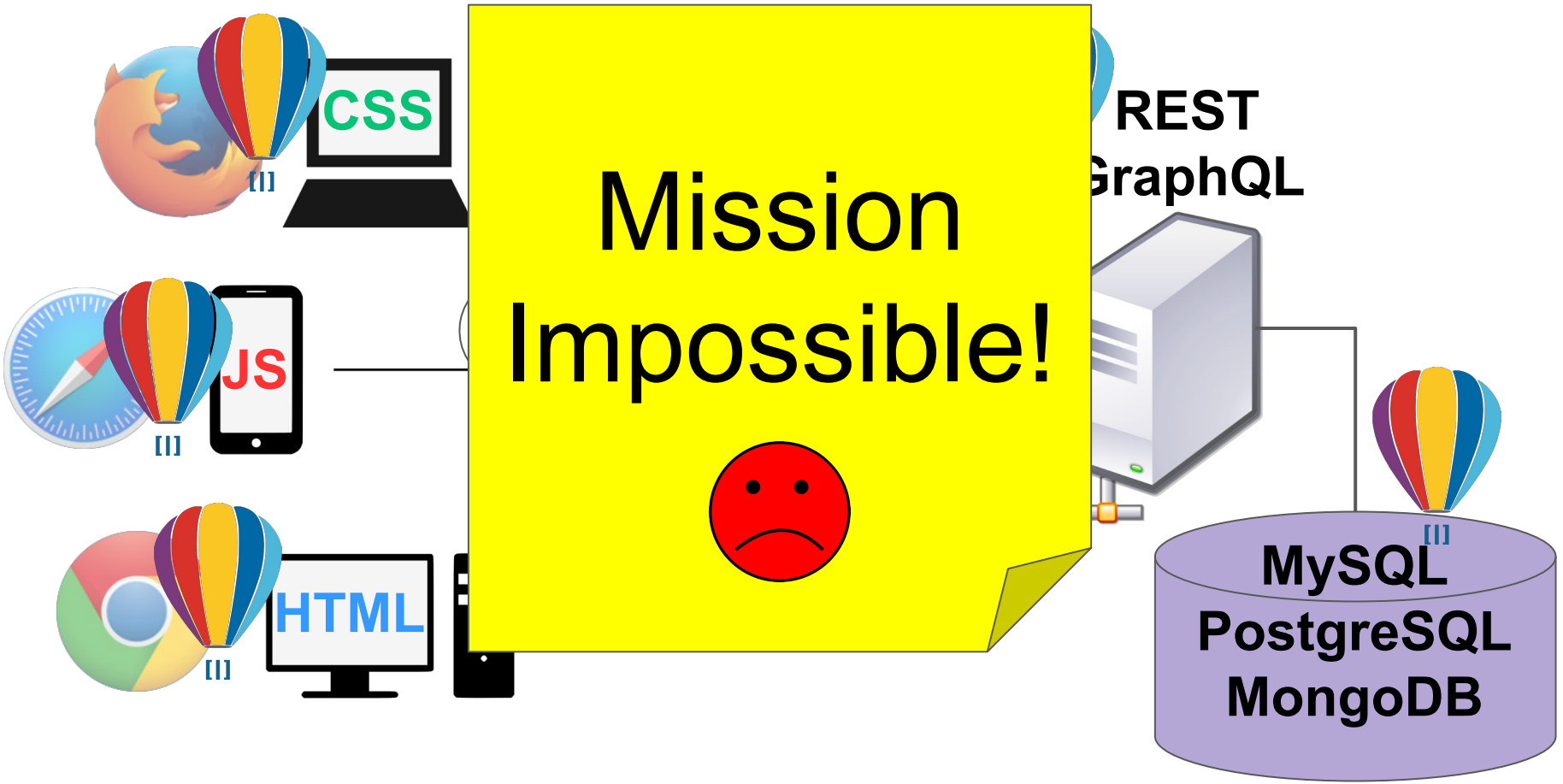

web language:JavaScript


web

Web Github Repositories



Smalltalk for the Web



Mission Impossible is ALWAYS Possible

Good artists copy;
Great artists steal

– Pablo Picasso

Good developers copy;
Great developers steal

– me ;-)

Reimplement

Good developers **copy**;
Great developers **steal**

– me ;-)

Reuse

Smalltalk for the Web

In a project with

Smalltalk +

Web Technologies?

Bridging Smalltalk

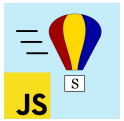
with

JavaScript & HTML & CSS

4 Approaches to Bridge Smalltalk with the Web

CodeParadise

PharJS

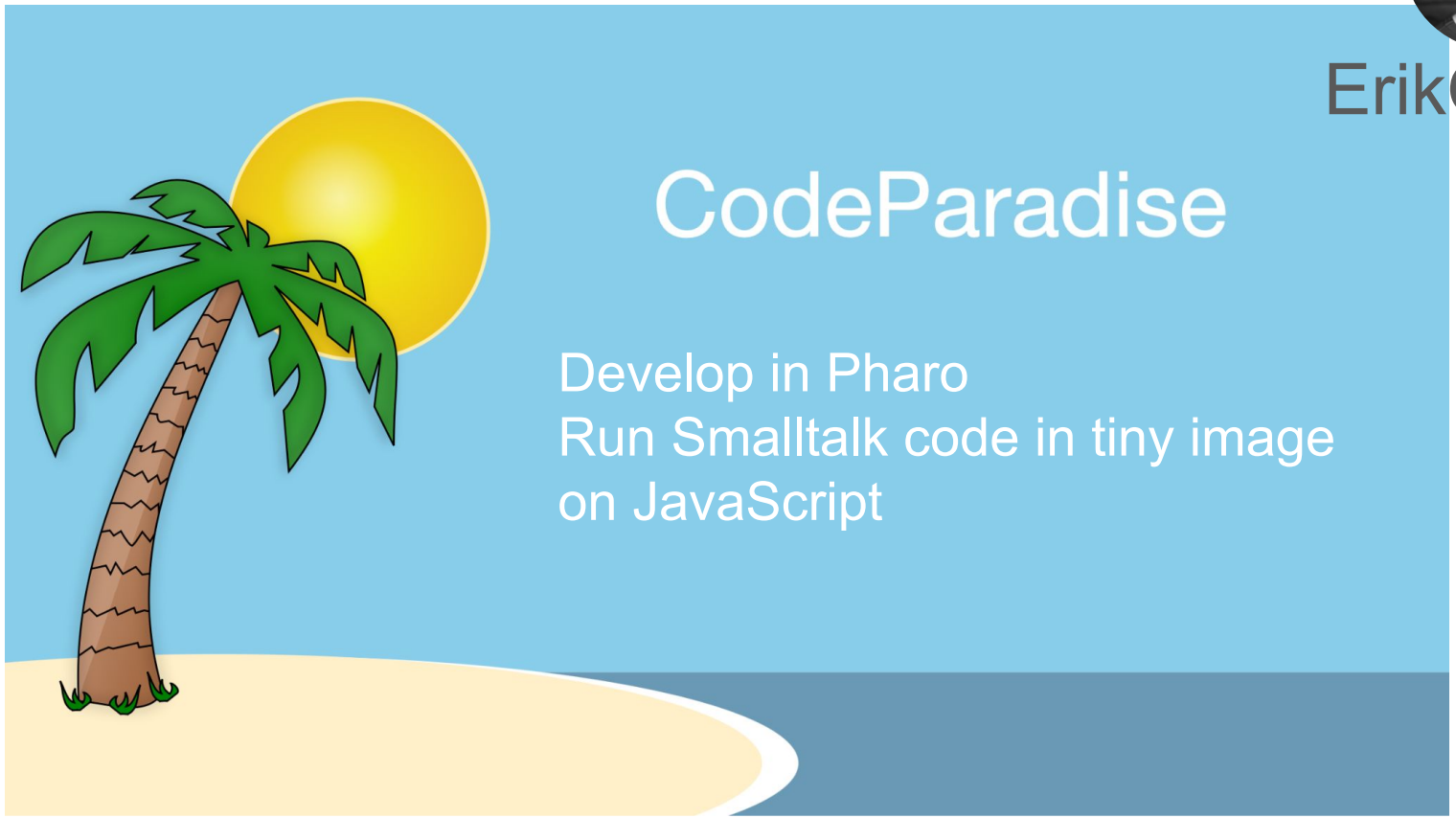


Small JS

WebST

CodeParadise

ErikOnBike

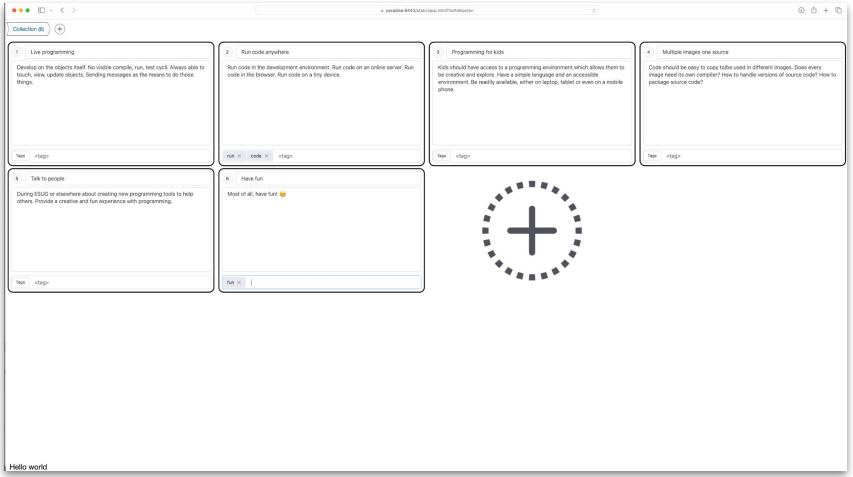


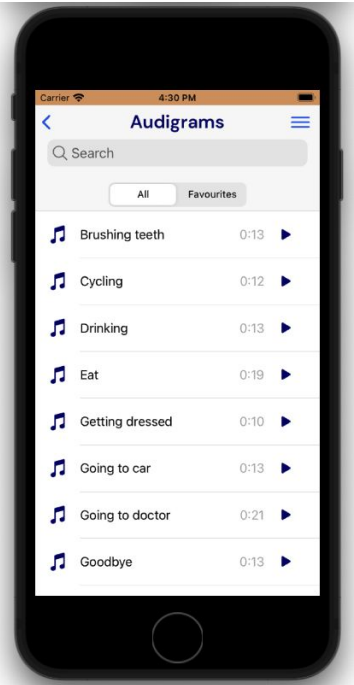
CodeParadise

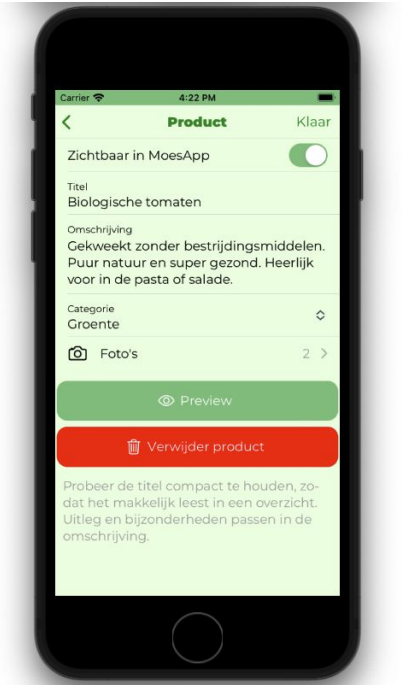
- What makes CodeParadise special?
  - No transpilation: run tiny image on JS based VM (SqueakJS)
  - Interactive development: update code without reload
  - All Smalltalk goodyness: Context, Process, non-local returns
- When you ❤️❤️❤️ Smalltalk and want/need to build for JS
- Start with Web or Node app and fire up your browser/node
- ⚠️ as addictive as Smalltalk itself
- ⚠️ webapps are most stable, but great potential for other apps (not everything is built yet for these other type of apps)

Longer demo: Friday July 4th 9:30 - Canyon (mobile apps)

CodeParadise - apps







webapps, mobile apps, node-based apps, cli apps…

…as long as it runs in a JavaScript environment

CodeParadise - JavaScript integration

- Tight coupling DOM & WebComponents and DOM events
- Wrap (any) other JavaScript classes
- Support for Promises (no distinction between sync/async)
- Possible to call Smalltalk from JavaScript (but mostly unused)
- Supported frameworks/libraries:

Browser
- Shoelace (partial)
- Ionic
- ChartJS
- Browser APIs (all partial)
  - Speech API
  - Media API
  - Geolocation API
  - Canvas API

Node.js
- sharp (image processing)
- better-sqlite3
- Nodemailer
- Node APIs (all partial)
  - Crypto API
  - FileSystem API
  - Net & HTTP API
  - Stream API

Mobile (PWA)
- Ionic
- (everything browser)

Nativize using Capacitor (similar to Cordova)


PharJS
Develop in Pharo
Run on JavaScript
Noury
Dave

Pharo 100%

Development → Production

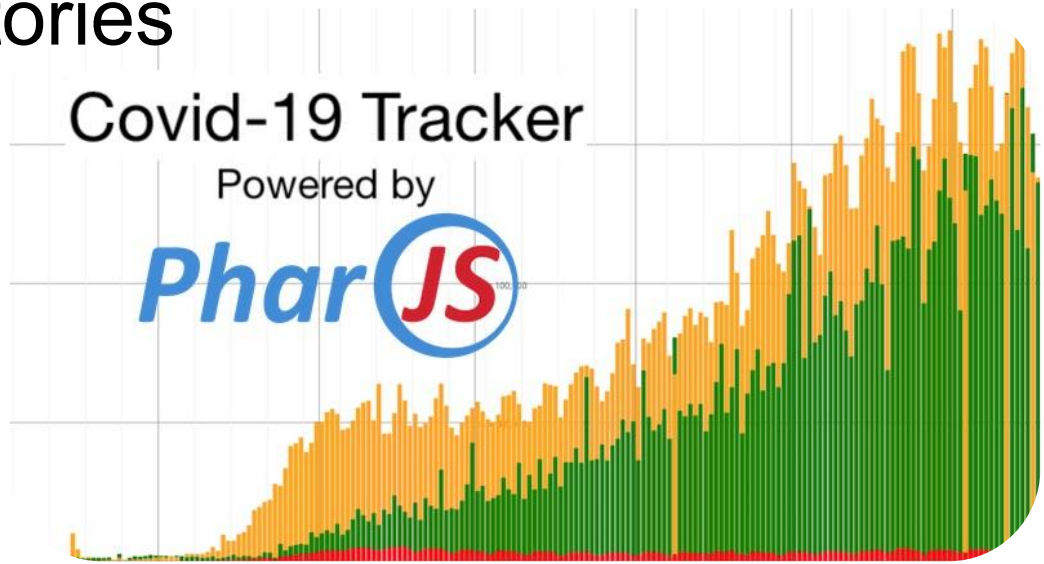
100% Javascript

Pharo 100%

1. Write Tests
2. Pass the tests
3. Export to JS

100% Javascript

PharJS Success Stories



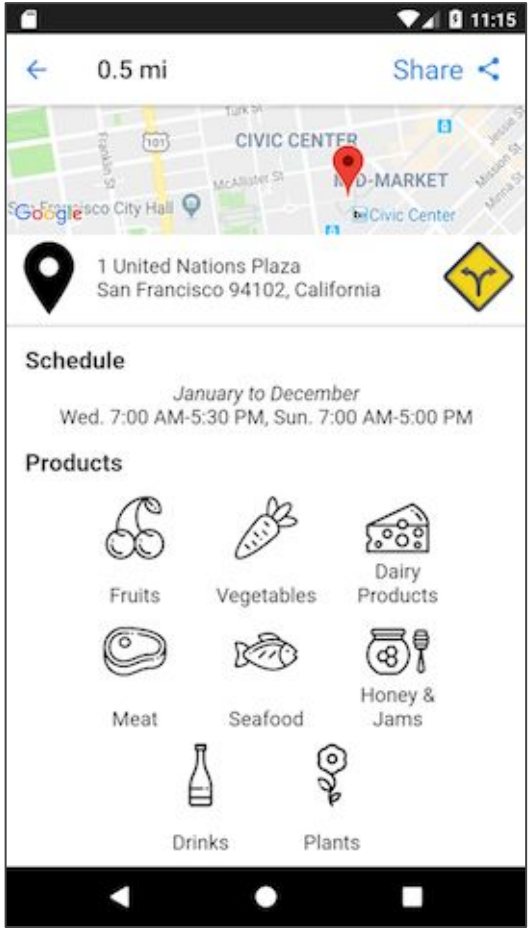
PharJS

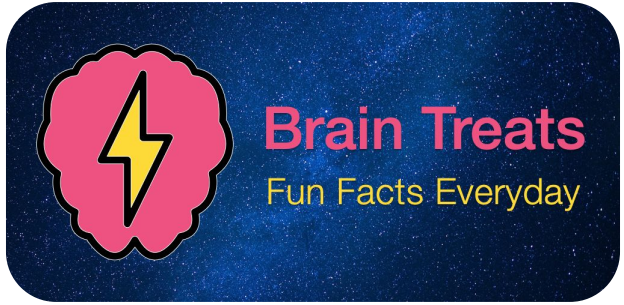
Smalltalk REPL

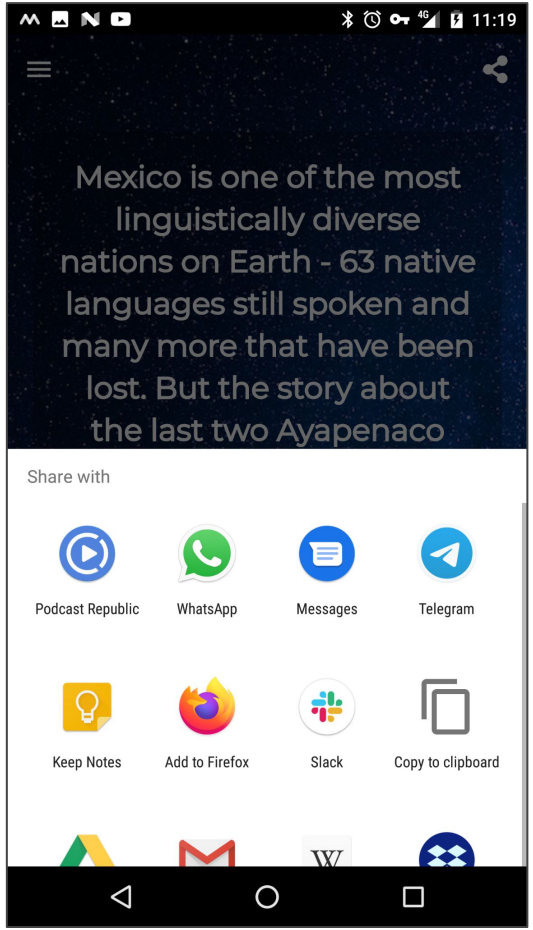
PLC3000

Teaching PLC Automation Made Easy

PharJS Success Stories







PharJS

10th Anniversary!

PharoJS

- What is it?
  - Transpiler: Converting Smalltalk to JavaScript
  - Support reuse JS libraries in Smalltalk during development
  - Test framework: Run Smalltalk tests that check JS behavior
  - Playground: Control JS generated program from Smalltalk
- Target Audience ?
  - Web client developers
  - Back-end developers with JS servers (ExpressJS…)
  - Desktop applications with JS (Electron…)
  - iOS + Android Mobile apps based on multi-platform a JS/webview framework (Cordova…)
- How to Use it?
  - Develop in Smalltalk
  - Test: Smalltalk test control Generated JS under test
  - Production: Export standalone JS
- When to Use it?
  - Reuse existing JS libraries
  - Build JS libraries for JS developers
  - Missing Smalltalk Runtime = Platform without Smalltalk VM port
  - Need for performance = JS Runtime is faster than the SmalltaLk one

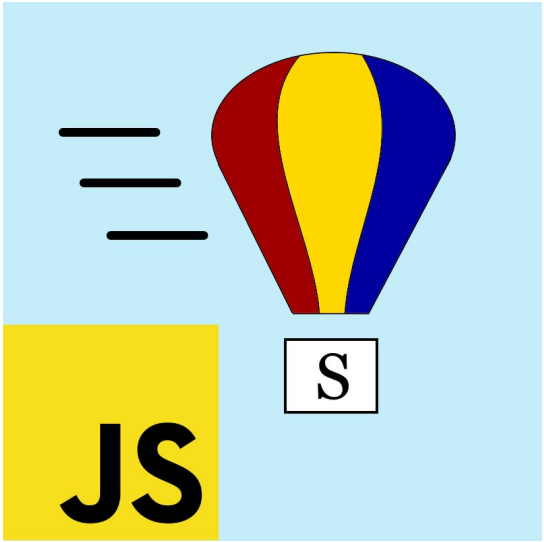
SmallJS

Smalltalk ***within*** the web

by:
Richard Ronteltap

for:
ESUG July 2025 - Gdansk

esug
[!] EUROPEAN SMALLTALK USER GROUP



Website: small-js.org

What is SmallJS?

- Transpiler from Smalltalk to efficient, lightweight JavaScript that runs in modern browsers ***and*** in Node.js.
- Full Smalltalk-80 language syntax support.
- Expansive ST library (image) for browser and Node support.
  - Seamless integration with HTML/CSS and ExpressJS.
- Development in the powerful Visual Studio Code IDE
  - With syntax coloring and step debugging!
- Built-in unit and GUI testing (browser).
- Example apps to get started quickly.
- Playground available (local or on small-js.org).

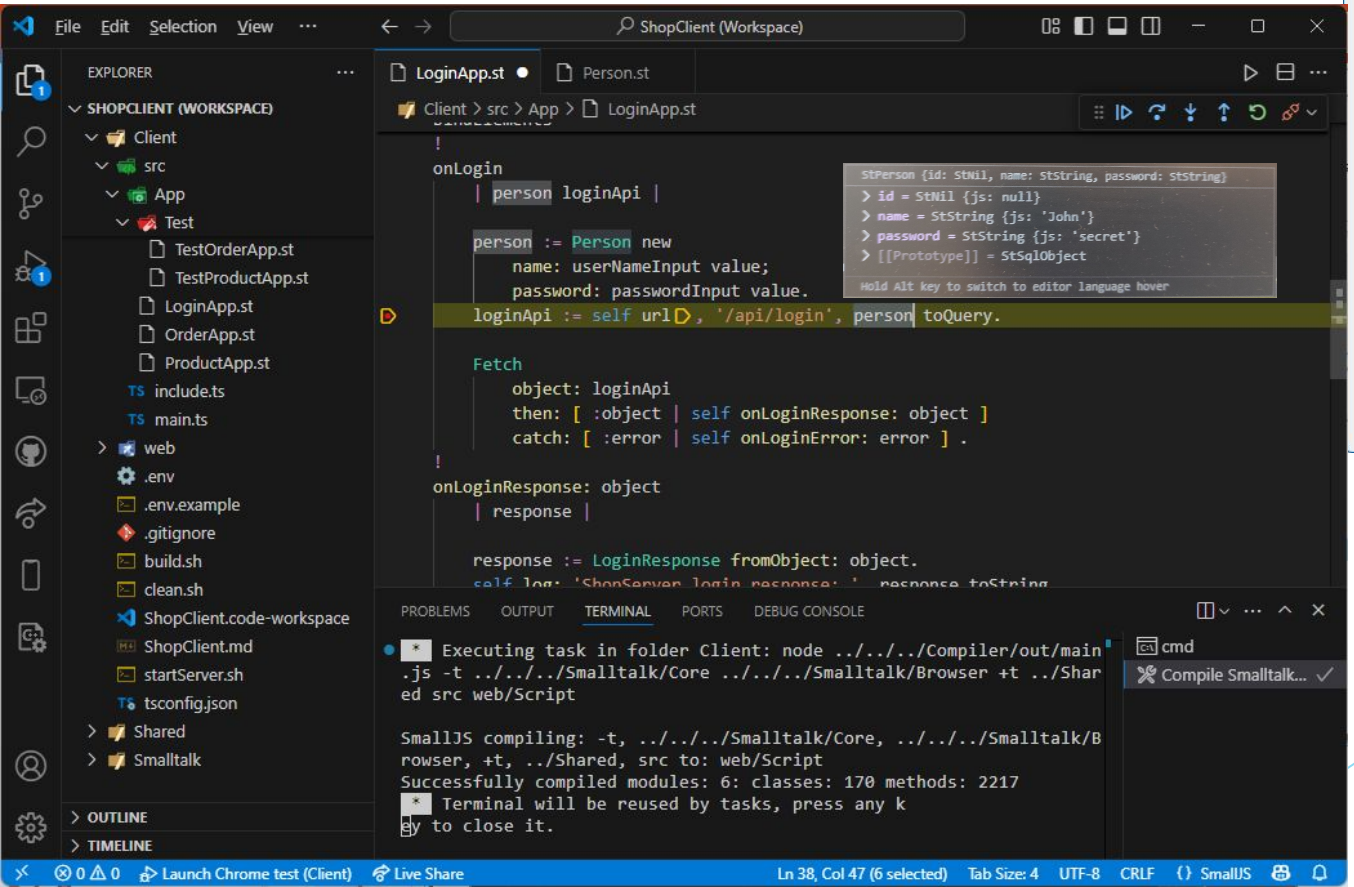
Target audience

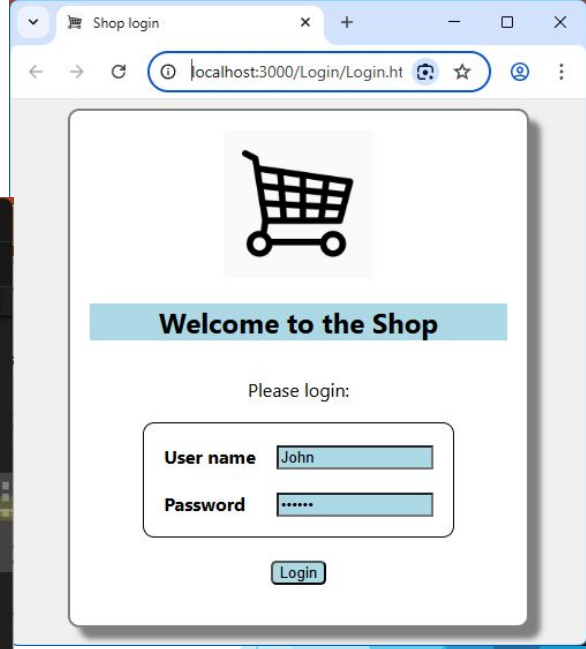
Developers that want to:

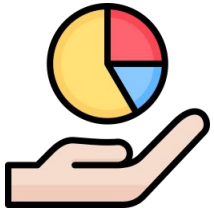
- ✔ Develop modern, complex web apps in Smalltalk.
- ✔ Use the *same* language for front-end and back-end.
- ✔ Have nicer, cleaner alternative for JS/TS.
- ✔ Have seamless integration with JS frameworks.
  - ✔ With full control and easily augmented.
- ✔ Use the friendly and powerful *file-based* VSCode IDE.
- ✔ Have source level step-debugging in all environments.
  - ✔ With access to underlying JS when needed


SAMPLE

How to use it?


File Edit Selection View
ShopClient (Workspace)
EXPLORER
SHOPCLIENT (WORKSPACE)
Client
src
App
Test
TestOrderApp.st
TestProductApp.st
LoginApp.st
OrderApp.st
ProductApp.st
include.ts
main.ts
web
.env
.env.example
.gitignore
build.sh
clean.sh
ShopClient.code-workspace
ShopClient.md
startServer.sh
tsconfig.json
Shared
Smalltalk
OUTLINE
TIMELINE
LoginApp.st
Person.st
Client > src > App > LoginApp.st
onLogin
| person loginApi |
person := Person new
name: userNameInput value;
password: passwordInput value.
loginApi := self url, '/api/login', person toQuery.
Fetch
object: loginApi
then: [:object | self onLoginResponse: object]
catch: [:error | self onLoginError: error] .
onLoginResponse: object
| response |
response := LoginResponse fromObject: object.
StPerson {id: StNil, name: StString, password: StString}
> id = StNil {js: null}
> name = StString {js: 'John'}
> password = StString {js: 'secret'}
> [[Prototype]] = StSqlObject
Hold Alt key to switch to editor language hover
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
Executing task in folder Client: node ../../../Compiler/out/main.js -t ../../../Smalltalk/Core ../../../Smalltalk/Browser +t ../Shared src web/Script
SmallJS compiling: -t, ../../../Smalltalk/Core, ../../../Smalltalk/Browser, +t, ../Shared, src to: web/Script
Successfully compiled modules: 6: classes: 170 methods: 2217
Terminal will be reused by tasks, press any key to close it.
cmd
Compile Smalltalk...
Launch Chrome test (Client)
Live Share
Ln 38, Col 47 (6 selected)
Tab Size: 4
UTF-8
CRLF
SmallJS


Shop login
localhost:3000/Login/Login.ht
Welcome to the Shop
Please login:
User name
John
Password
Login



When to use it?

- Complex, rich, interactive websites.
- Where new JS libs need to be added quickly.
- Existing JS/TS projects for gradual migration.
  Leveraging your knowledge of JS libs.
- Use **one**, powerful, file-based IDE: VSCode
  Everything for your project in one place.
  Safely separated from your running code...
- Easy source code control with standard tools.
- Easy, minimal deployment, no stripping.
- Create multi-provider AI apps (new!).

Noury

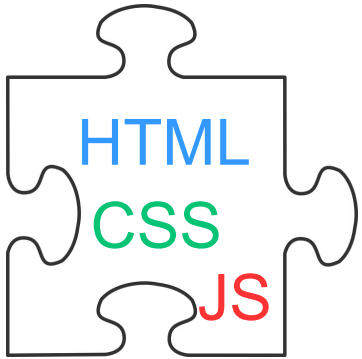
WebST

Web Components in Smalltalk

Web Components

W3C Standard

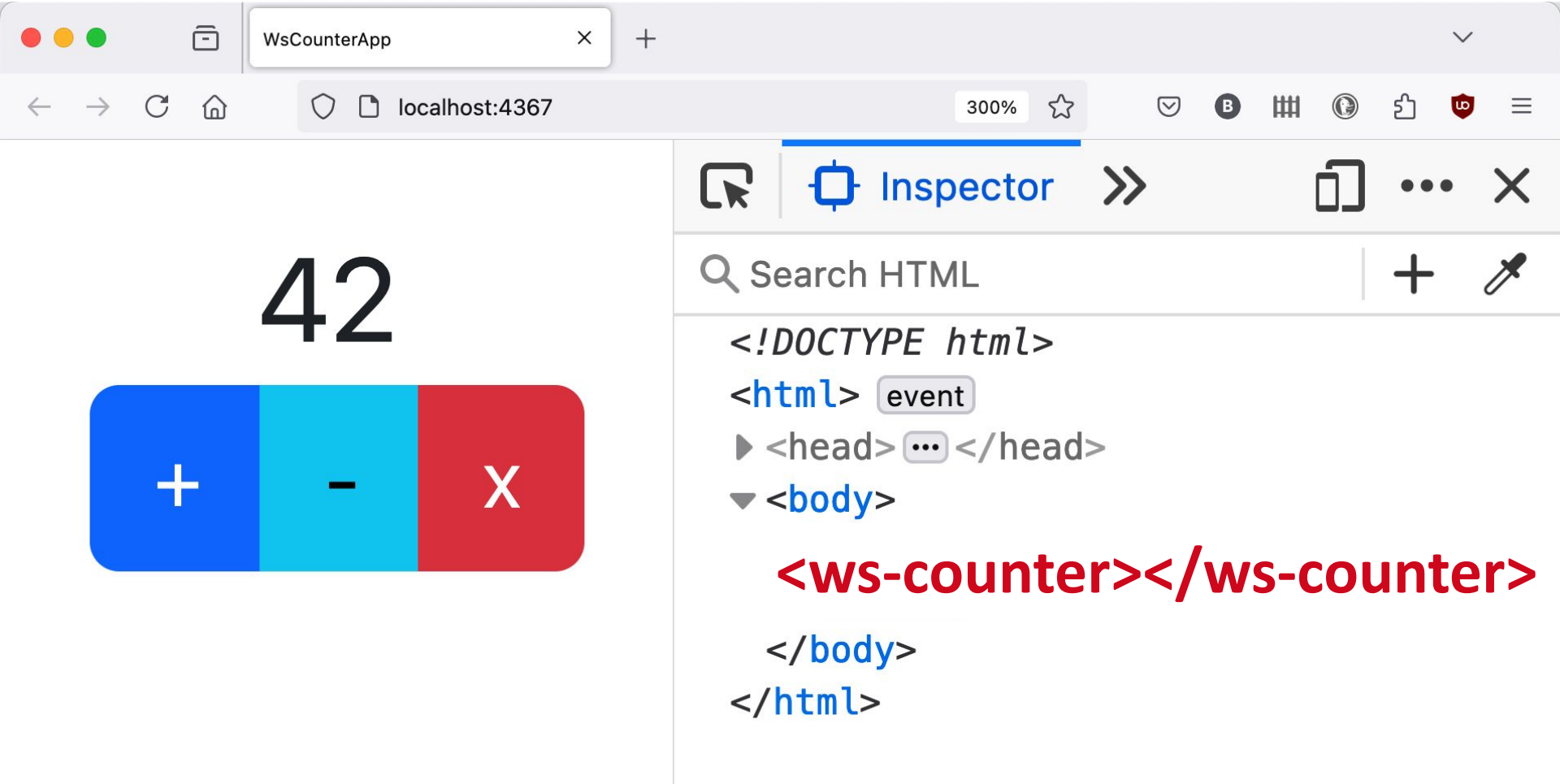
- Custom HTML tags
- Embed HTML + CSS + JavaScript
- Encapsulation = No collisions



Component

Web App UI

A Web Component Example



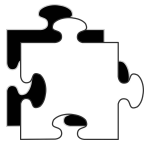
WebST = **Build** Components with **Smalltalk**

Smalltalk

Web Component

WebST = **Reuse** Components with **Smalltalk**

Third-part Web Components



Smalltalk

Standalone Web Client

Pharo 100%

1. Write tests
2. Pass the tests
3. Export

Server + Clients 100%

HTML + JavaScript + CSS

WebST

- What is it?
  - Framework for building Web Components in Smalltalk = Generates HTML + JavaScript
  - Framework to build Websites with Multiple HTML pages based on Web Components in Smalltalk
  - SUnit extension to support testing Components and Web Sites
- Target Audience ?
  - Front-end UI developers
  - UI component libraries developers
  - Website developers
- How to Use it?
  - Develop Components and Pages in Smalltalk, possibly Reuse Existing JS, CSS libraries
  - Run in Smalltalk tests on HTML and JS exported from Smalltalk code of Components and pages
  - Export Standalone HTML + JS
- When to Use it?
  - Build Websites, mobile apps (Cordova…), desktop apps (Electron…)
  - Build Web component libraries

WebST Relies on PharoJS + Seaside

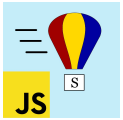
Summary & Comparison of the 4 Approaches

4 Approaches to Bridge Smalltalk with the Web

CodeParadise: Run tiny Pharo Images on JavaScript

PharoJS: Develop in Pharo, run on JavaScript

SmallJS: New Smalltalk flavour with seamless JavaScript integration in browsers and Node.js, enabling file-based development from Visual Studio Code.



WebST: Develop in Pharo, run Web Components = HTML + JS + CSS

Comparison Criteria

- Files & Entities
  - JS, HTML, CSS
  - Smalltalk

Virtual Machine

Image

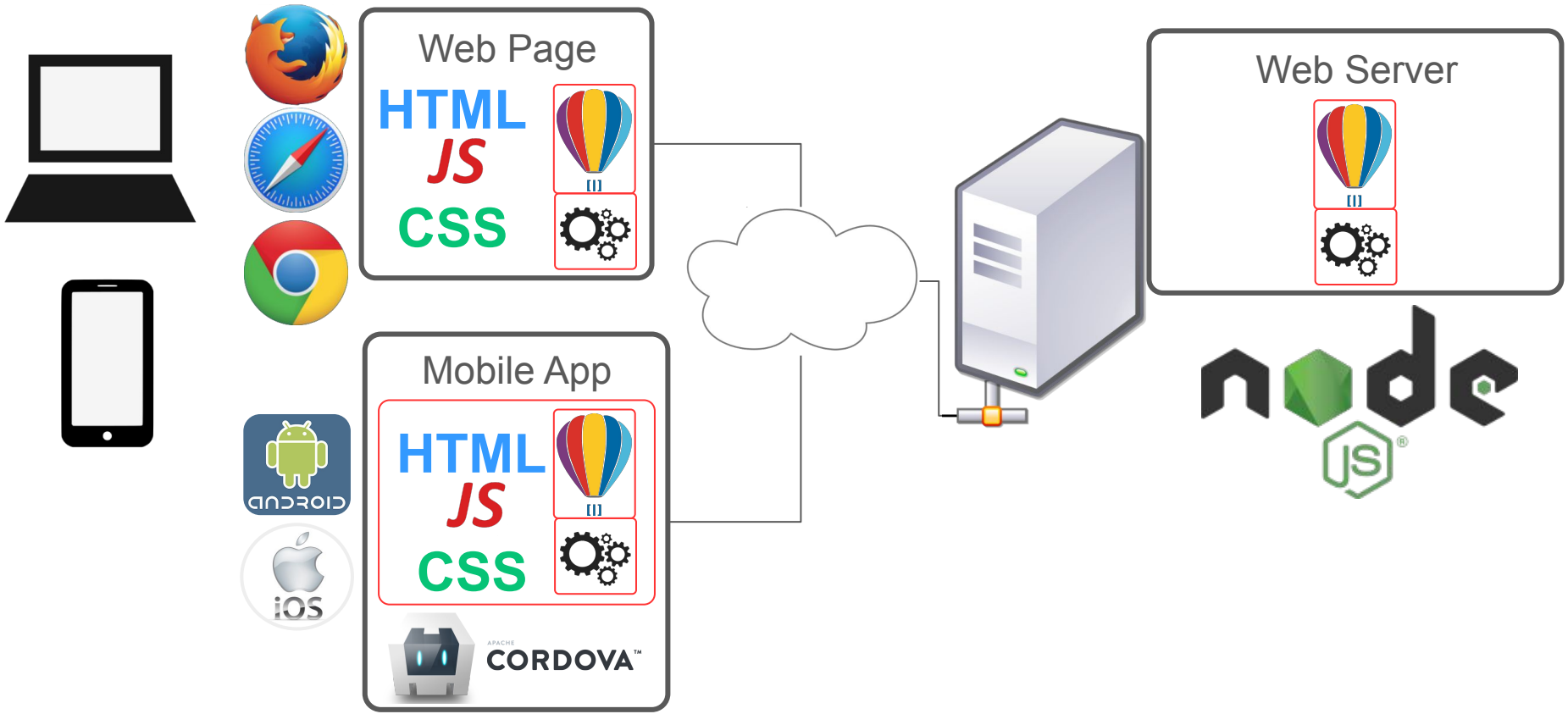
Source

- Targets

Clients

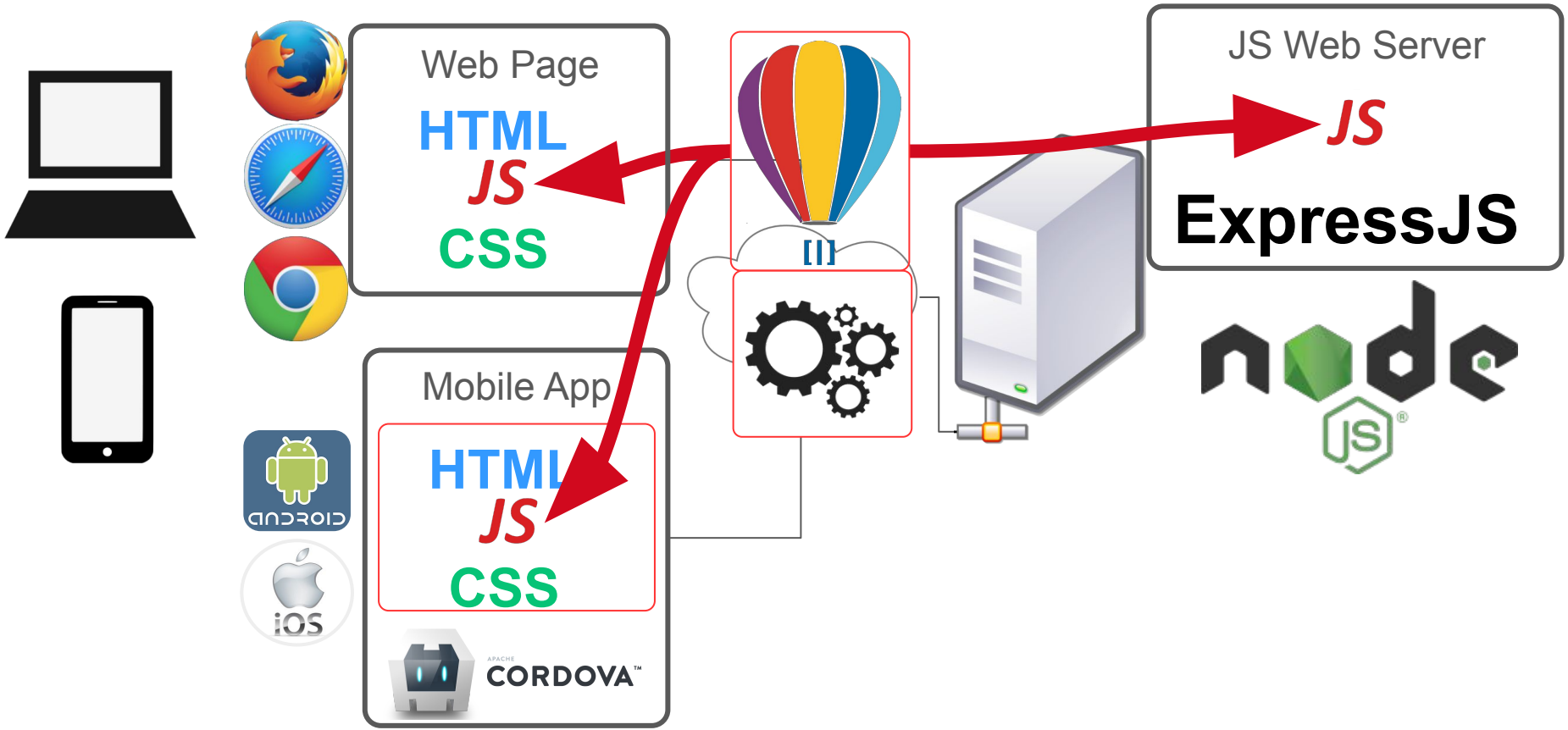
Server

CodeParadise: Run tiny Smalltalk Images on JS

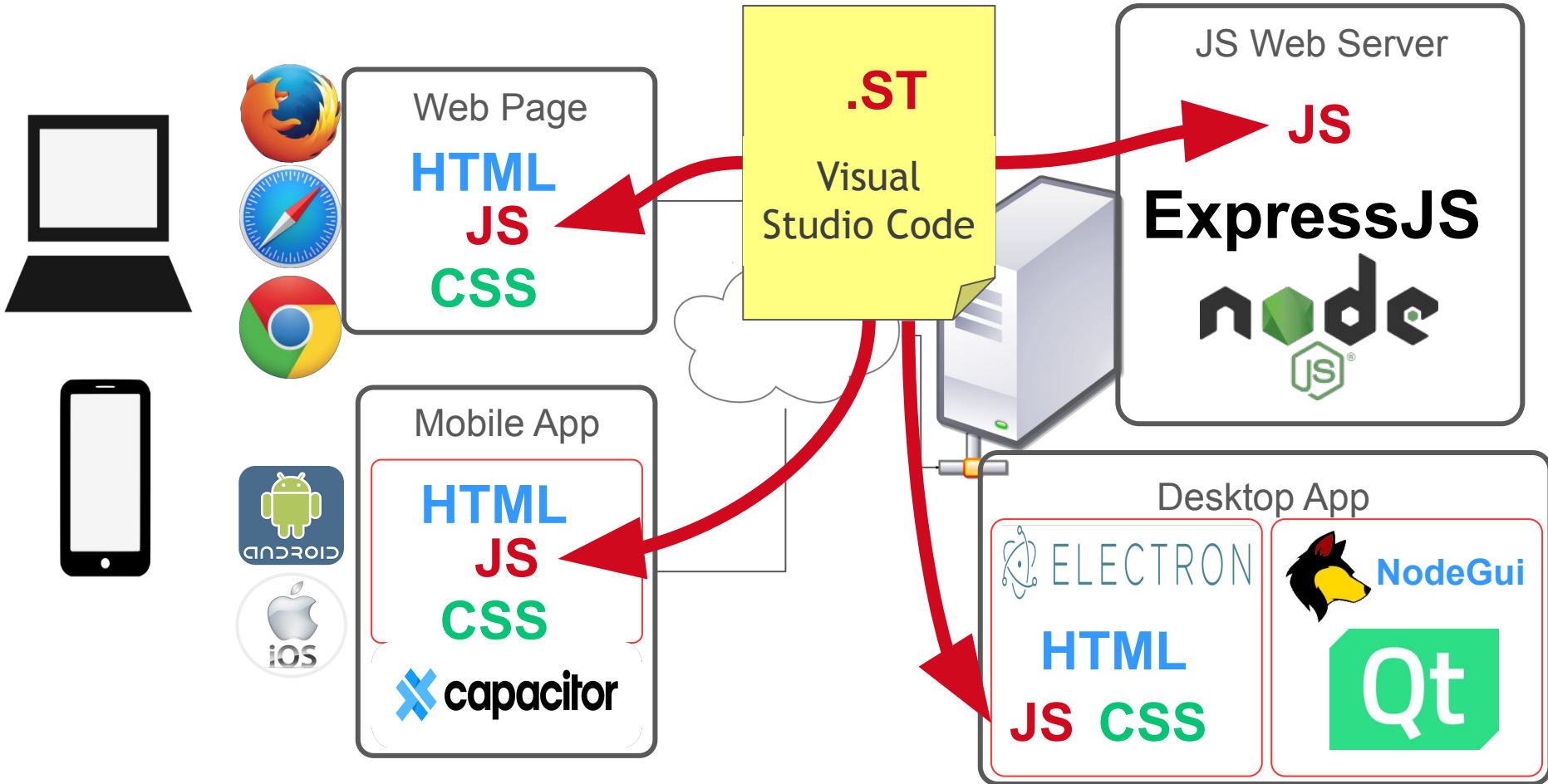


PharJS Develop in Pharo, Run on JavaScript

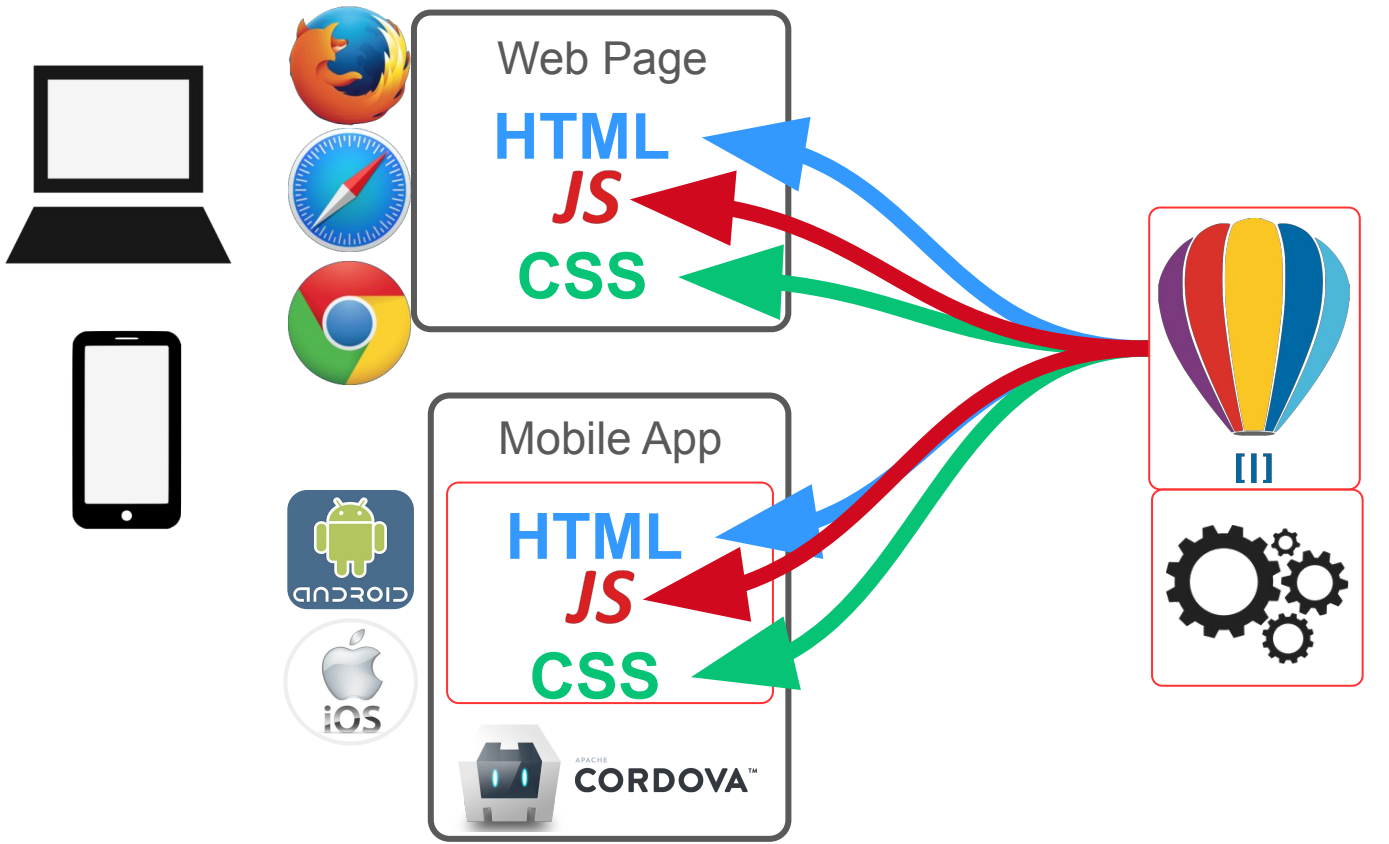
Test & Export from the Image



SmallJS: New Smalltalk flavour for JavaScript



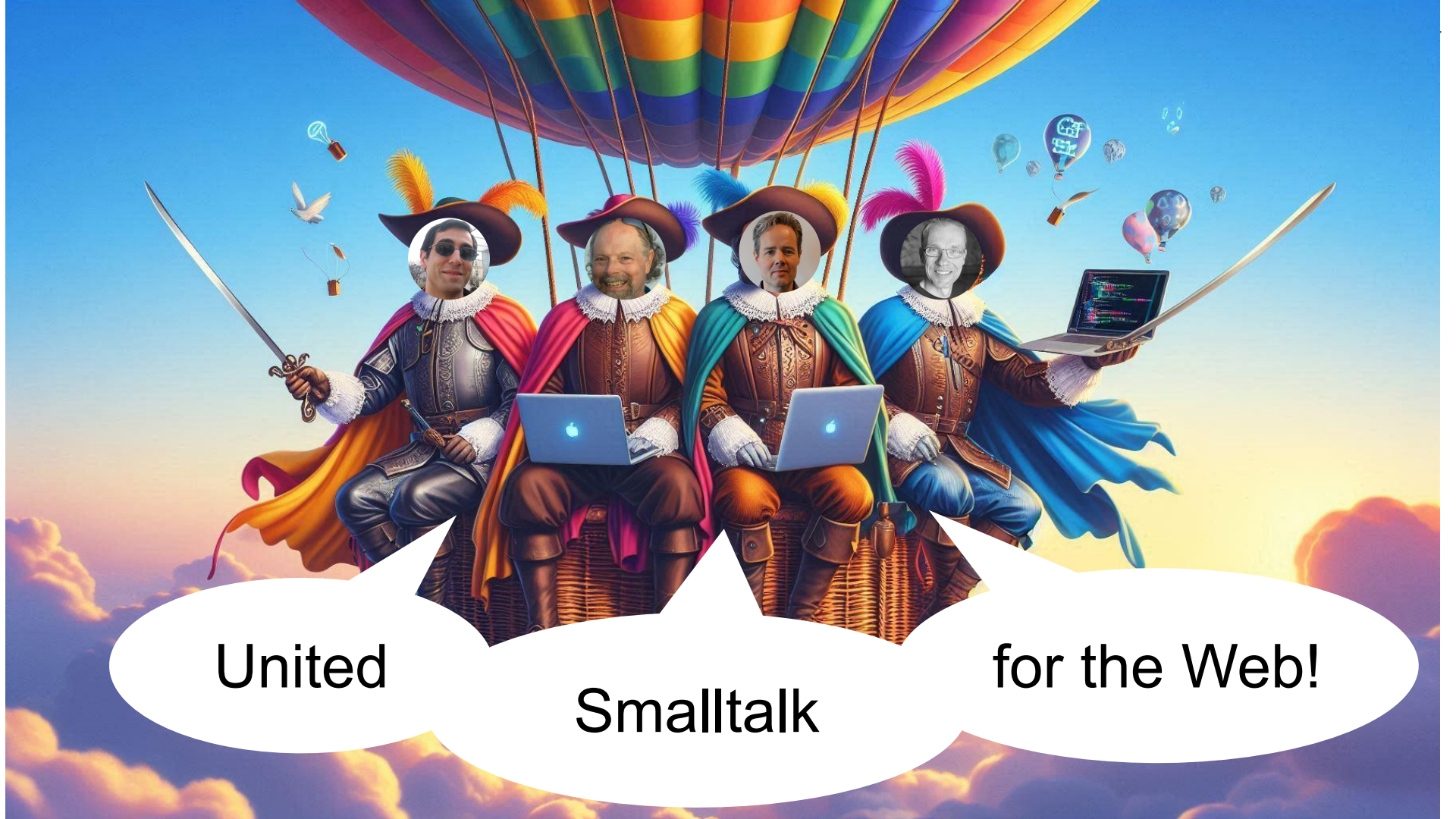
WebST: Develop in Pharo, Run Web Components



Smalltalk Web Initiative


Smalltalk!
All the Time!
Everywhere!

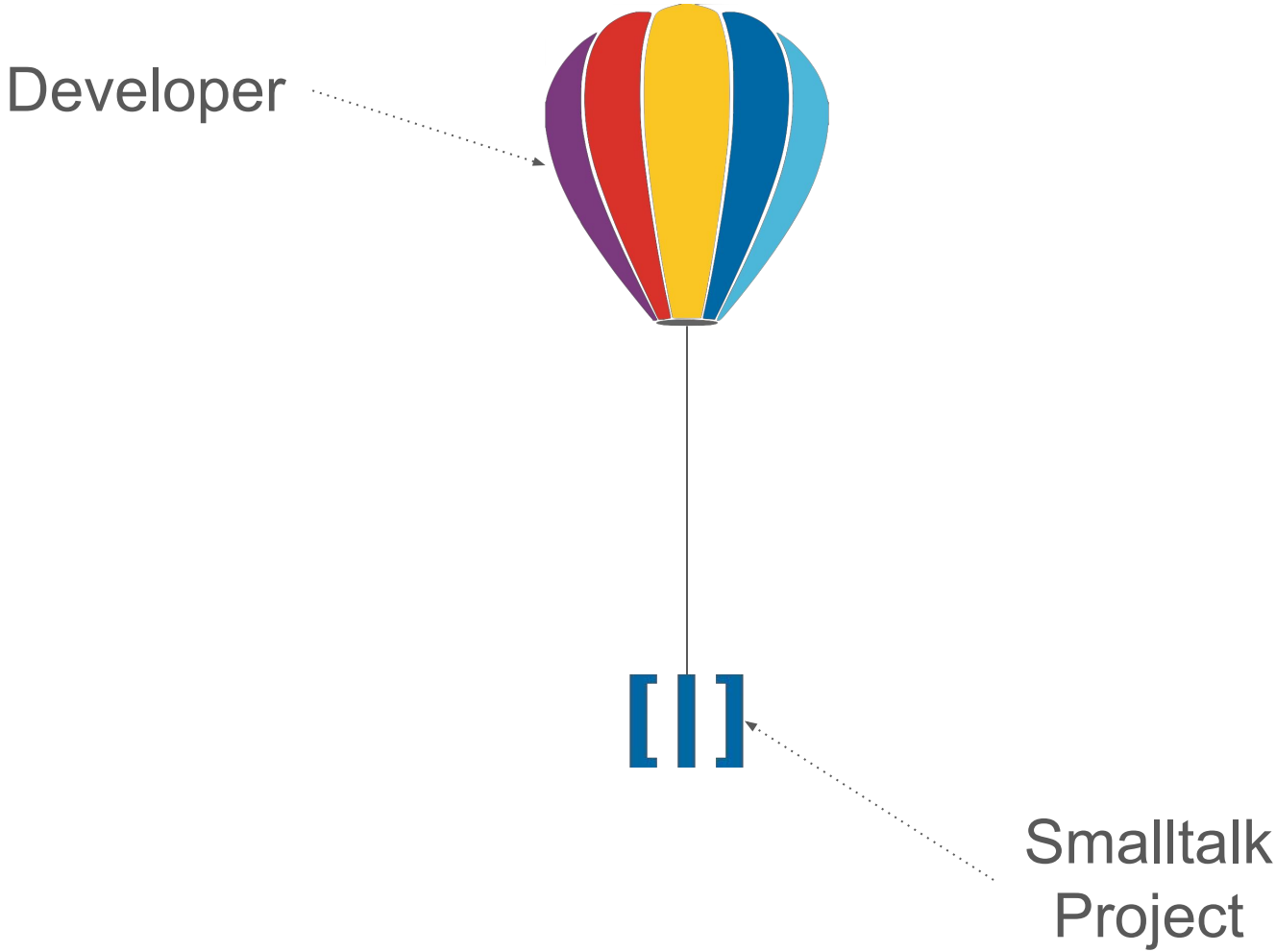

Let's Compare our Solutions!
What Can We Share?
When to Use Them?
Areas of Overlap ?
When are They Used?
Differences?
How are They Used?
Target Audience?


United
Smalltalk
for the Web!

Goal of the Smalltalk Web Initiative

- **Unite** bridges to web technology
  - Encourage **collaboration**
  - Beyond : CodeParadise, PharoJS, SmallJS and WebST
- **Open** to all Smalltalk dialects
  - Cross-dialect reuse as in Seaside?
  - Grease?
- **Free** Open-source, MIT Like license
  - Sustainable
  - Business friendly

The  Factor


Developer
Smalltalk
Project

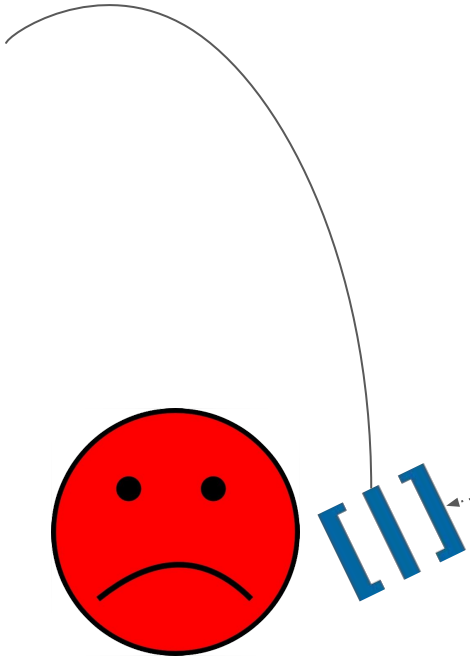
Developer

Balloon Factor == 1

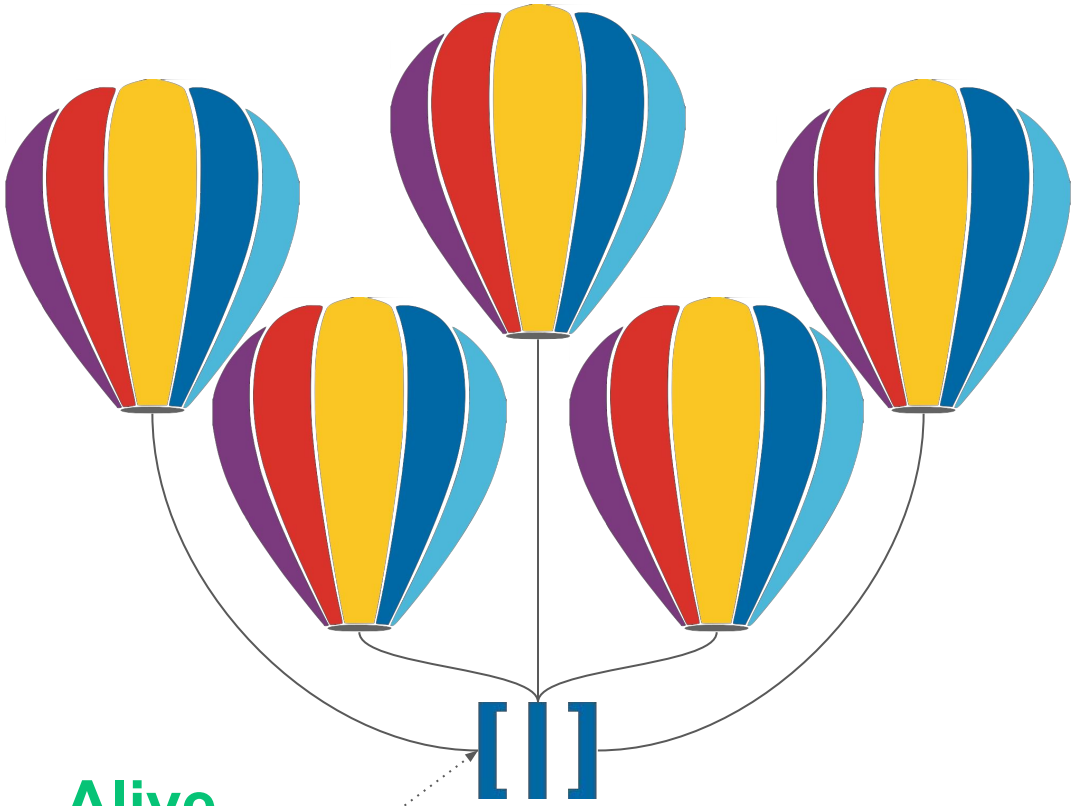
Alive
Smalltalk
Project

No
Developer

Balloon Factor == **0**

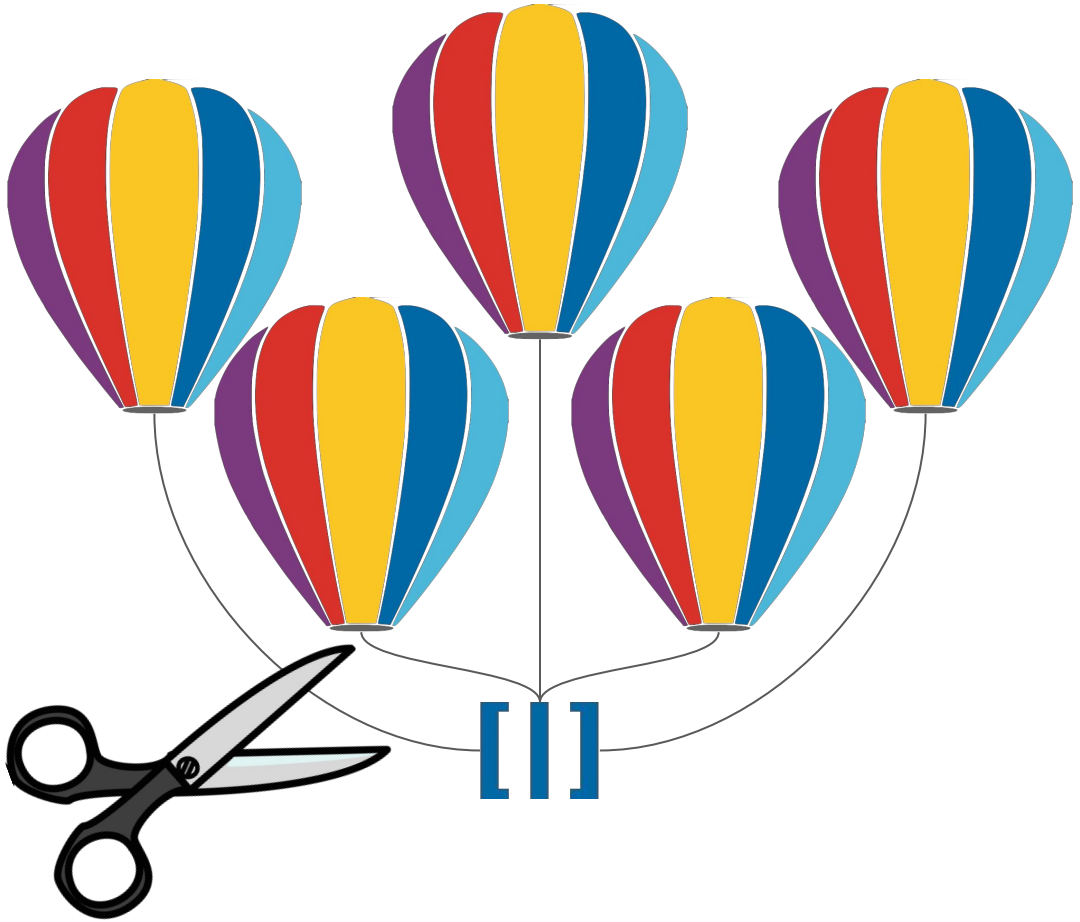


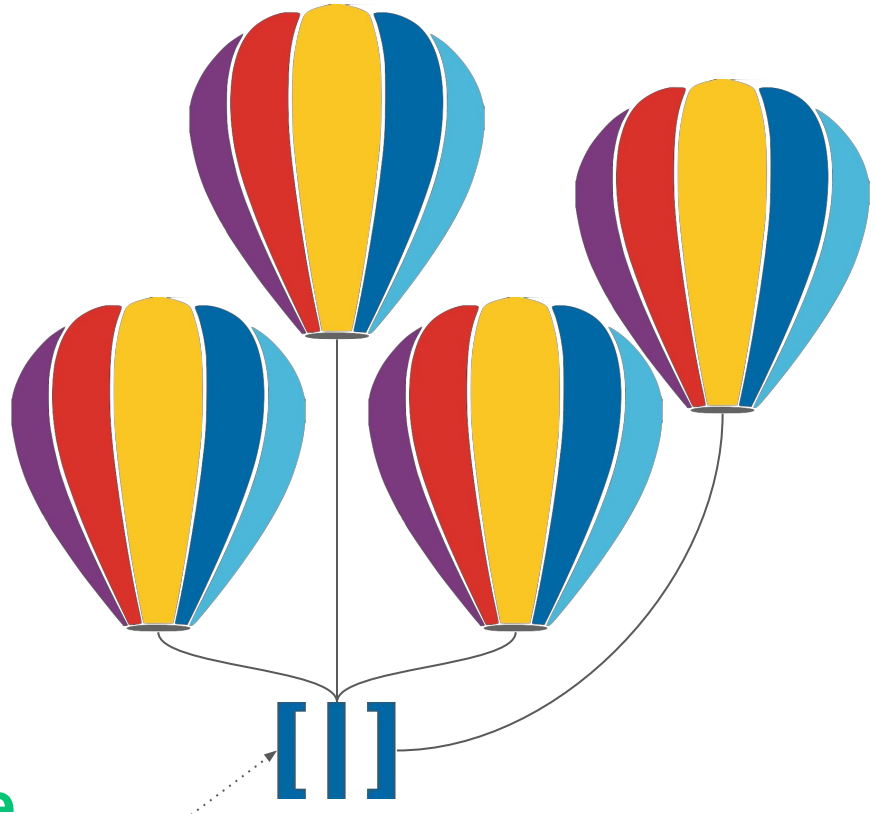
Dead
Smalltalk
Project



Alive
Smalltalk
Project

Balloon Factor == 5





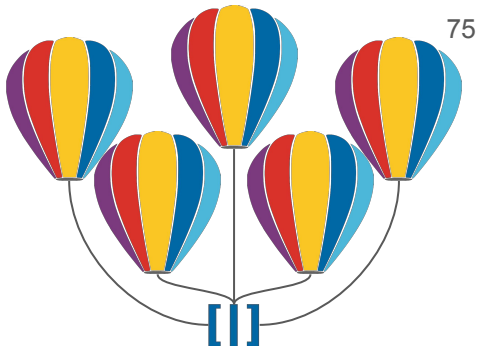
Alive
Smalltalk
Project

Balloon Factor == 4

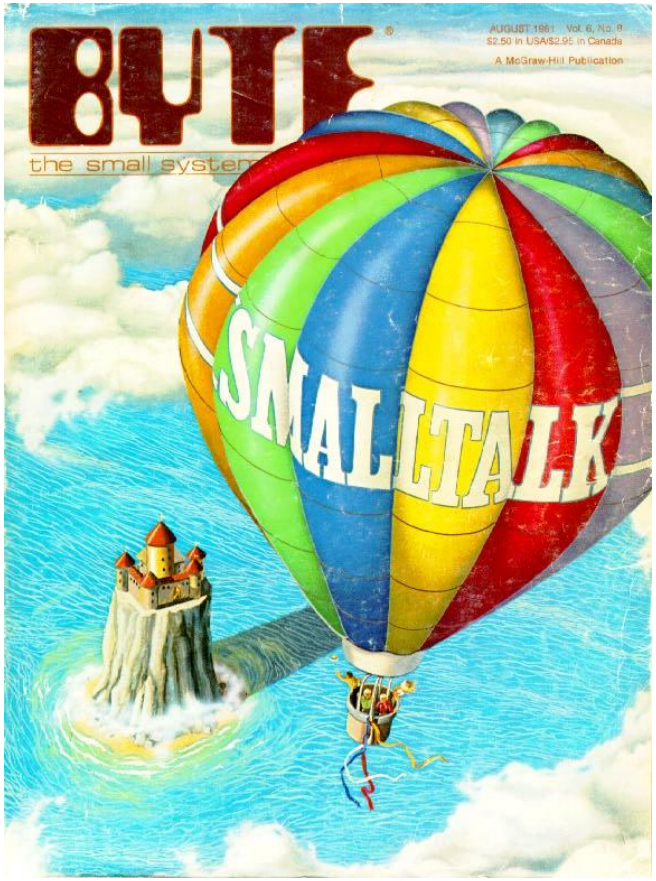
We need High Balloon Factors!

But, we are a Small Community!

Strategy for High Balloon Factors!

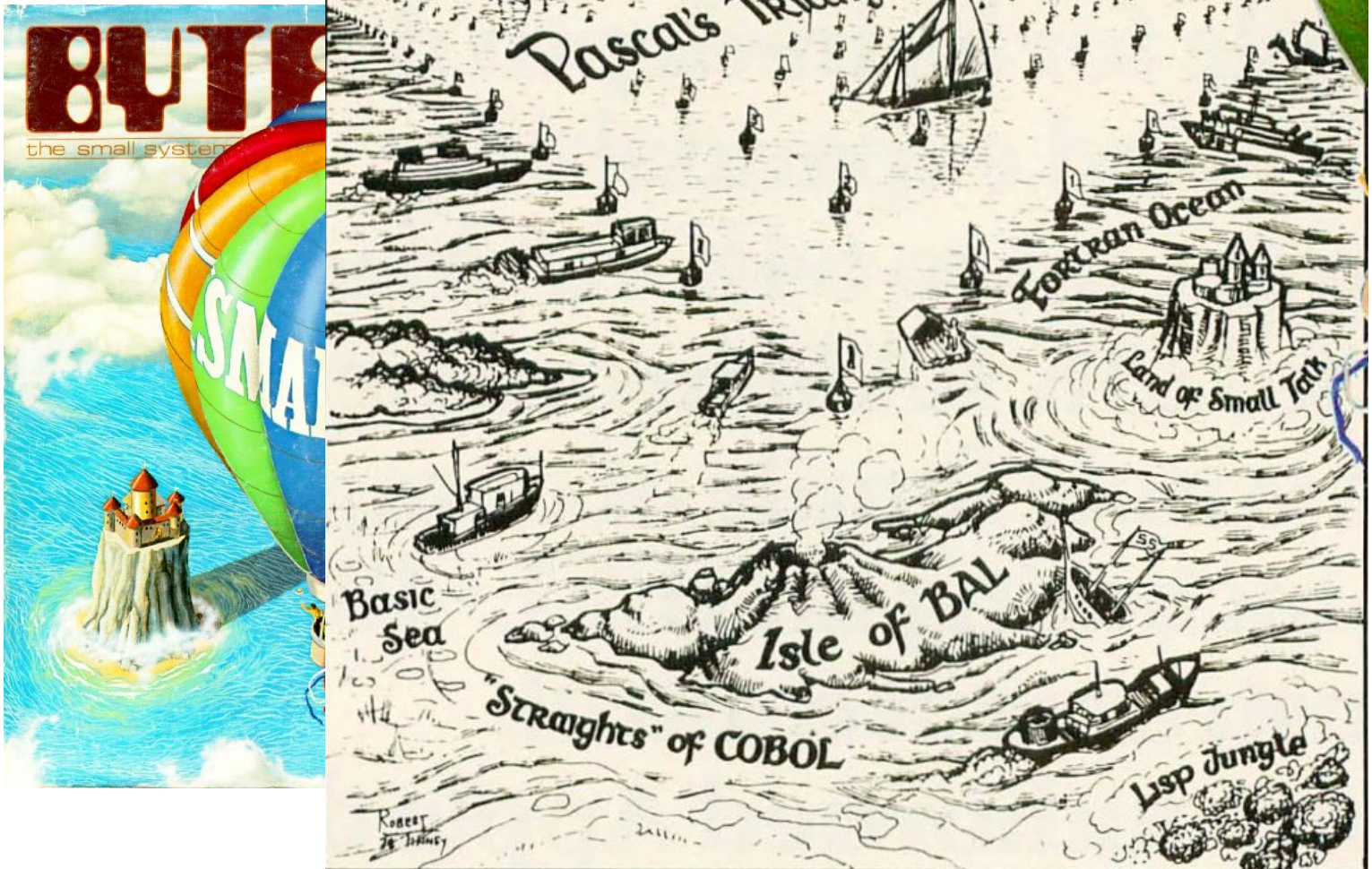


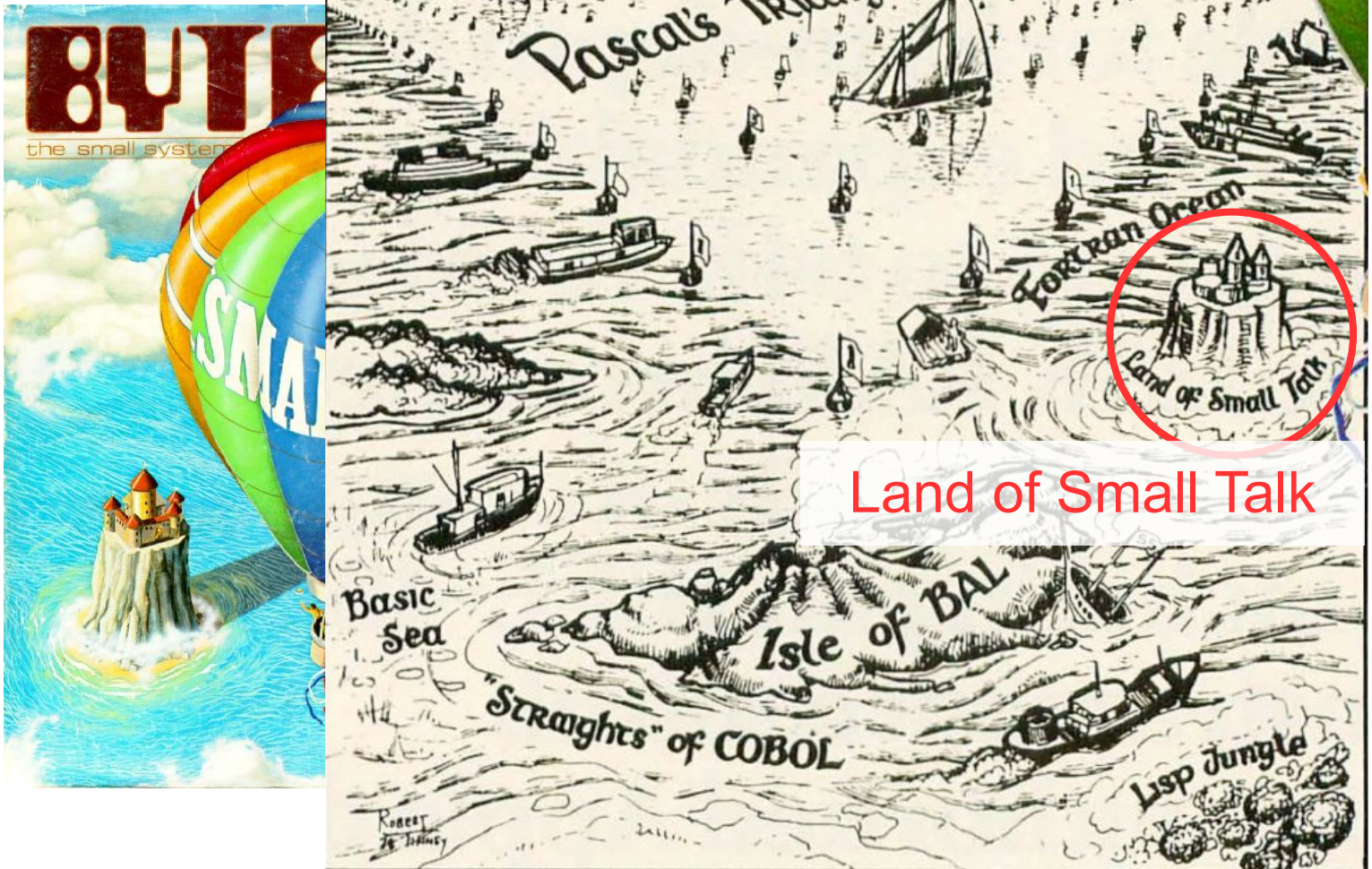
- Unite around **Fewer** but **Critical** projects
  - **Leverage other technologies**
  - Make ST libraries **reusable by other technologies**
- Free Open-Source with permissive **MIT-Like License**
- **Cross-dialect projects** *à la* Seaside/Grease



Why is the

Smalltalk Symbol a Hot Air Balloon ?


BYTE
the small system
Pascal's Triangle
Fortran Ocean
Land of Small Talk
Basic Sea
Isle of BAL
"Straights" of COBOL
Lisp Jungle



Isolated Island!

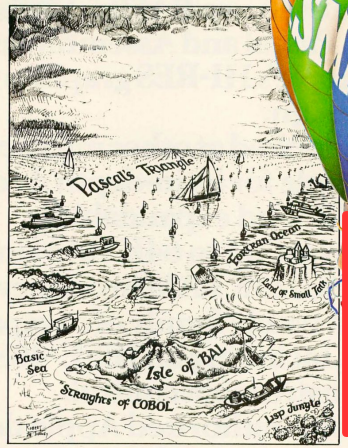
https://www.worldradiohistory.com/Byte_Magazine.htm

Introducing the Smalltalk-80 System

Adele Goldberg
Manager, Learning Research Group
Xerox Palo Alto Research Center
3333 Coyote Hill Rd
Palo Alto CA 94304

It is rare when one can indulge in one's prejudices with relative impunity, poking a bit of good humored fun to make a point.

With this statement, Carl Helmers opened his remarks in the "About the Cover" section of the August 1978 issue of BYTE. The issue was a special on the language Pascal, so Helmers took the opportunity to present Pascal's triangle as drawn by artist Robert Tinney. The primary allegory of the cover was the inversion of the Bermuda Triangle myth to show smooth waters within the area labeled "Pascal's Triangle." In explaining the allegory, Helmers guided the traveler through the FORTRAN Ocean, the BASIC Sea, around the Isle of BAL, and up to the Land of Smalltalk.



Traveling upward (in the picture) through heavy seas we come to the pinnacle, a snow white island rising like an ivory tower out of the surrounding shark infested waters. Here we find the fantastic kingdom of Smalltalk, where great and magical things happen. But alas . . . the craggy aloofness of the kingdom of Smalltalk keeps it out of the mainstream of things.

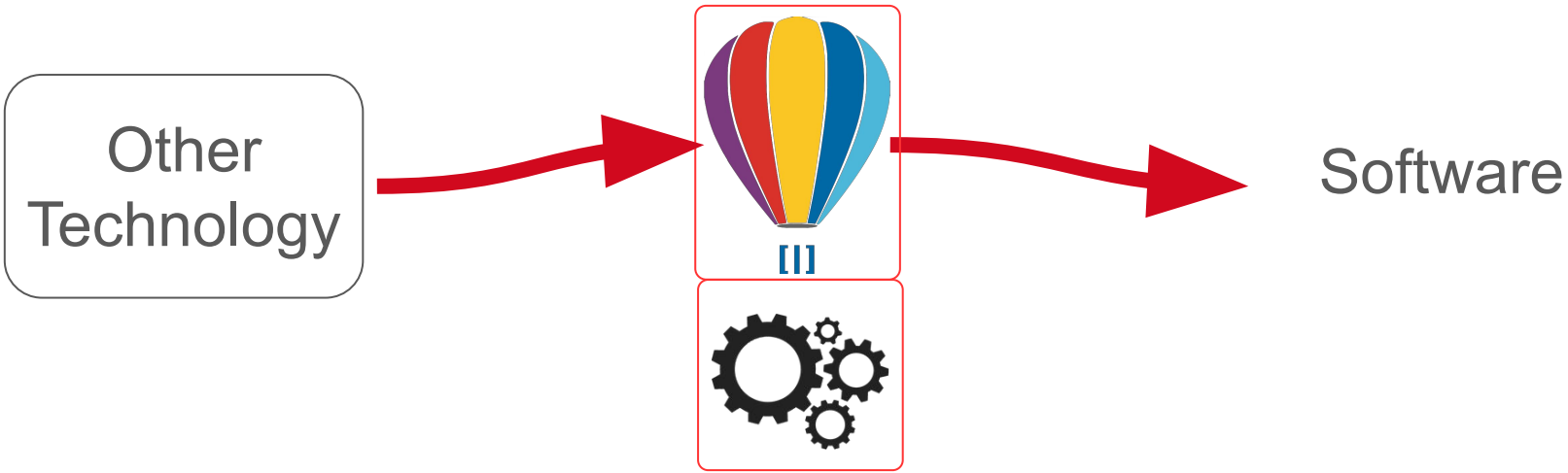
It is rare when one can indulge in one's fantasies to respond to so pointed a remark as that provided by the then editor of BYTE. This month's cover design presents just such an opportunity. It depicts the clouds clearing from around the kingdom of Smalltalk, and, with banners streaming, the Smalltalk system is taking flight into the mainstream of the computer programming community. This cover was also executed by Robert Tinney, to the delight of the Learning Research Group (LRG) of the Xerox Palo Alto Research Center. LRG is the group that has designed, implemented, and evaluated several generations of Smalltalk over the past ten years.

The balloon on the cover symbolizes the Smalltalk-80 system that is being released this year for more general access. The release is in the form of publications and a file containing the Smalltalk-80 programming system. Twelve articles describing the system appear in this issue of BYTE. Through such publication, LRG's research will become generally accessible, dispelling the clouds.

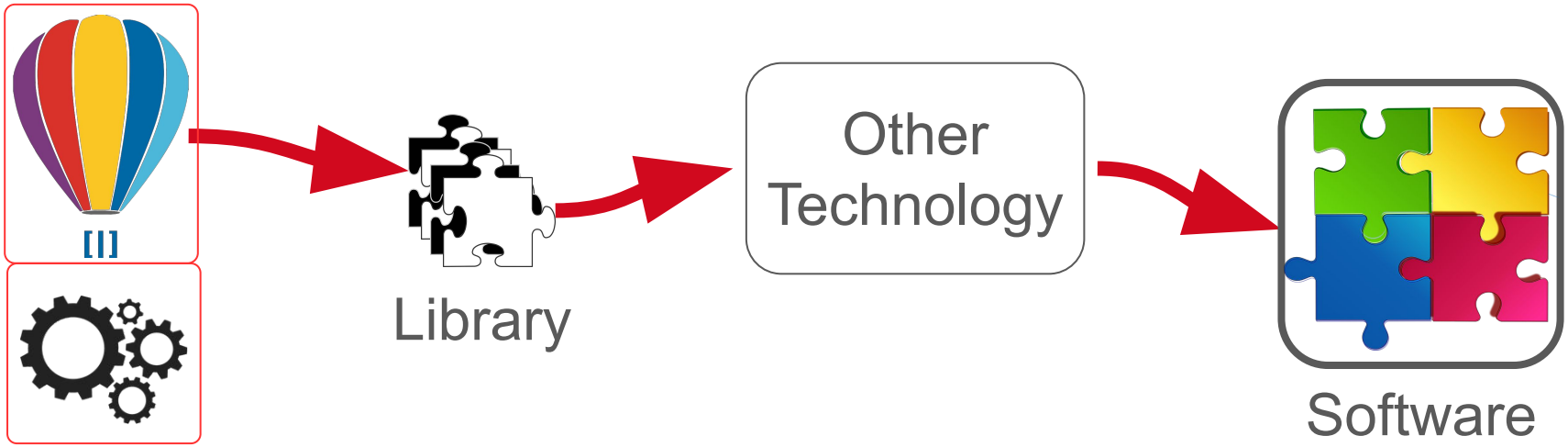
Smalltalk is the name LRG assigned to the software

the **Smalltalk** system is **taking flight into** the **mainstream** of the computer **programming community**.

Leverage Other Technology



Smalltalk Librairies Reusable by Others



Join the SmalltalkWeb Musketeers!

- Come see us at the break
- Talk to us over the Pharo Discord

#web

- Make PRs on the web pages

SmalltalkWeb.github.io